



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1458  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/765,717

01/26/2004

Shaz Qadeer

3382-66931

2565

26119 7590 06/25/2007  
KLARQUIST SPARKMAN LLP  
121 S.W. SALMON STREET  
SUITE 1600  
PORTLAND, OR 97204

EXAMINER

DENG, ANNA CHEN

ART UNIT

PAPER NUMBER

2191

MAIL DATE

DELIVERY MODE

06/25/2007

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

**Office Action Summary**

Application No.

10/765,717

Applicant(s)

QADEER ET AL.

Examiner

Anna Deng

Art Unit

2191

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 13 April 2007.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-20 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-20 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- |  |   |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892)   | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                       | 5) <input type="checkbox"/> Notice of Informal Patent Application                       |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____  |

**DETAILED ACTION**

1. This action is in response to amendment filed on 4/13/2007.
2. Claims 1-13, and 18-20 have been amended.
3. Claims 1-20 are pending.
4. The rejection under 35 U.S.C. 101 to claims 1-20 is withdrawn in view of applicant's amendment.
5. The rejection under 35 U.S.C. 112 to claims 1-20 is withdrawn in view of applicant's amendment.
6. Claims 1-20 stand finally rejected under 35 U.S.C. 102 (b) as being anticipated by Christiaens, US 2002/0120428 A1.

***Response to Amendment***

***Claim Rejections - 35 USC § 102***

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

8. Claims 1-20 are rejected under 35 U.S.C. 102 (b) as being anticipated by Christiaens, US 2002/0120428 A1 (hereinafter Christiaens).

**Per Claim 1 (Currently Amended):**

Christiaens discloses:

- A computer comprising one or more computer-readable media and a processor, the computer-readable media containing instructions ([0038]), which, when executed by the processor on the computer, cause the computer to perform that actions of a system for detecting a data race in a concurrent program ([0018], lines 1-6), the system comprising:

- a program sequentializer module configured to accept a concurrent program as input ([0034], "a computer language compiler mechanism for converting a multithreaded source program described by a program language into a computer executable machine language from a computer system"; also, Figs. 17a-17b, [0246]-[0247], [0255]-[0261], the sequentializer module consists of elements in 17a or 17b), and create as output a sequential program having assertions ([0255], "the actual race detection is carried. For this, 20 bytecodes, for instance, are instrumented which read or write to an object. Each time such a byte code is executed, it is checked whether it is a global object...If a global object is being dealt with, the extra data structures can be accessed and it can be verified, ...whether this instruction represents a data race. If so, this is flagged to the user (emphasis added)");
- wherein, when the sequential program is analyzed by a program analyzer, the assertions cause an error message to be produced by the program analyzer when the concurrent program contains a data race (Fig. 1, 111 and 109, [0102], lines 23-27, [0106]).

**Per Claim 2 (Currently Amended):**

Christiaens discloses:

- The computer of claim 1, wherein the system further comprises a sequential program analyzer module which analyzes the sequential program and produces error messages if assertions are not met and wherein the assertions are created to be checked by the sequential program analyzer ("a report of potential data races", [0025]).

**Per Claim 3 (Currently Amended):**

Christiaens discloses:

- The computer of claim 1, wherein the sequential program utilizes a single runtime stack (Fig. 13, stack 1501, [0192]) and the program sequentializer module is further configured to add

data structures (Fig. 4, instrumentation 404, a thread information structure 410, [0169]-[0172]) to the received code, the added data structures at least comprising:

- a multiset of thread pointers which comprises pointers to threads which have been created but have not yet been scheduled on the runtime stack (Fig. 4, pointers 405-406, 408, [0114]-[0115], and [0117]); and
- a global boolean exception variable which, when set, causes the sequential program to remove from the runtime stack the currently-executing thread (Figs. 4-5 and 13, TID field, [0116], and [0183]-[0189]).

**Per Claim 4 (Currently Amended):**

Christiaens discloses:

- The computer of claim 3, wherein: the multiset of thread pointers is limited to a maximum number of pointers; and the added data structures further comprise a global multiset size variable, which indicates the maximum number of pointers (the length of the translation table 611, Fig. 6, [0136]-[0137]).

**Per Claim 5:**

Christiaes discloses:

- The computer of claim 3, wherein: the program sequentializer module is further configured to receive an indication of a target variable which will be analyzed for data races (Fig. 13, global set 1508, [0019], [0183]); and
- the added data structures further comprise a global access variable which indicates, for the analyzed target variable, the current type of access being requested of the analyzed target variable ([0036], and [0038]).

**Per Claim 6:**

Christiaens discloses:

- The computer of claim 5, wherein the program sequentializer module is further configured to insert instrumentation into the received code (Fig. 4, instrumentation 404, thread information structure 410), the instrumentation at least comprising:
- a scheduling function which selects a thread pointer from the multiset and schedules the thread indicated by the pointer on the runtime stack (synchronization operation, [0171]);
- an exception macro which sets the global boolean exception variable and causes an executing thread to be removed from the runtime stack ( $S_{tot}$ , [0194]-[0195]);
- a read-checking function, which checks to see that the global access variable does not indicate that the target variable is being written to and then sets the global access variable to indicate that the target variable is being read from (Fig. 14, 1609, 1010, 'read information structure', [0205]); and
- a write-checking function, which checks to see that the global access variable does not indicate that the target variable is being read from or written to and then sets the global access variable to indicate that the target variable is being written to (Fig. 14, 1610, 1618, 'write information structure', [0206]).

**Per Claim 7 (Currently Amended):**

Christiaens discloses:

Art Unit: 2191

- The computer of claim 6, wherein the read-checking and write-checking functions contain assertions about the global access variable which can be checked by the sequential program analyzer module (Fig. 14, 1611 and 1618, [0207]-[0218]).

**Per Claim 8 (Currently Amended):**

Christiaens discloses:

- The computer of claim 6, wherein the instrumentation is inserted so that it will execute nondeterministically in the sequential program (Fig. 4, instrumentation 404, [0170]-[0174]).

**Per Claim 9 (Currently Amended):**

Christiaens discloses:

- The system of claim 1, wherein the sequential program output by the sequentializer module is in the form of source code (sequential execution order of code, [1056], [0162]).

**Per Claim 10 (Currently Amended):**

Christiaens discloses:

- The computer of claim 1, wherein the sequential program output by the sequentializer module is in the form of an abstract syntax tree or a control-flow graph (Fig. 7, sequential execution order of events in one thread, T<sub>1</sub>, [0162]).

**Per Claim 11 (Currently Amended):**

Christiaens discloses:

- The computer of claim 1, wherein the concurrent program received by the sequential analyzer is in the form of source code (Figs 8-10, T<sub>1</sub> and T<sub>2</sub>, [0163]-[0166]).

**Per Claim 12 (Currently Amended):**

Christiaens discloses:

- The system of claim 1, wherein the concurrent program received by the sequential analyzer is in the form of an abstract syntax tree or control-flow graph (Figs. 8-10, and 13, [0174]-[0175]).

**Per Claim 13 (Currently Amended):**

Christiaens discloses:

- A method of analyzing a concurrent program for data races ([0018], "a computer implemented method for detecting inconsistent dynamic concurrency state transitions, especially data races in execution of multi-thread programs which are amenable to object reachability analysis"), the method comprising:
  - receiving a concurrent program ([0034]-[0035], "compiler mechanism for converting a multi-threaded source program described by a program language into a computer executable machine language ... receiving the source program");
  - receiving a least target variable to be analyzed for data race ([0106], "(read or write to a member variable are analysed to find inconsistent concurrency state transitions such as data races"); and
  - creating a sequential program from the concurrent program, the sequential program containing assertions such that, during an analysis of the sequential program, when assertions are not met, the analysis of the sequential program indicates the presence of a data race in the concurrent program for the target variable (Fig. 1, [0102], "The Java source code 101, once processed is intended to execute concurrently in a computer system 107 (CPU) as described above ... Java source code 101 can be compiled by a compiler 102... The function of the monitor 111 is to produce a report 109 on concurrency state information concerning concurrently executing threads. The report may contain information on data races



Art Unit: 2191

occurring while the class files 103 are executing in the system 107"; also, [0106], "for example, it two bytecodes modify a member variable of an object in the global set ... then a data race is reported", emphases added),

**Per Claims 14-17:**

These are method version of the claimed system discussed above (claims 1-8), wherein all claimed limitations also have been addressed and/or covered in cited areas as set forth above. Thus accordingly, these claims are also anticipated by Christiaens.

**Per Claim 18 (Currently Amended):**

This is computer-readable medium version of the claimed method discussed above (claim 13), wherein all claimed limitations also have been addressed and/or covered in cited areas as set forth above. Thus accordingly, this claim is also anticipated by Christiaens.

**Per Claims 19-20:**

These are computer-readable medium version of the claimed method discussed above (claims 13-15), wherein all claimed limitations also have been addressed and/or covered in cited areas as set forth above. Thus accordingly, these claims are also anticipated by Christiaens.

***Response to Arguments***

9. Applicant's arguments filed 4/13/2007 have been fully considered but they are not persuasive.

Applicant argued:

Claim 1

The cite sections of Christiaens do not describe, teach, or suggest "a program sequentializer module configured to accept a concurrent program as input and create as output a sequential program." In its rejection of the above-quoted language of claim 1, the Action cites of Figures 17a and 17b, as well

Art Unit: 2191

as paragraphs 0246-0247 and 0255-0261 of Christiaens. ... Parts of these sections describe instrumentation of objects ... and bytecode. ... but do not describe the creation of a sequential program from a concurrent program.

Examiner's responses:

Christiaens clearly teaches and suggests limitations "a program sequentializer module configured to accept a concurrent program as input and create as output a sequential program as recite in claim 1 rejection above, (Christiaen, [0034], "a computer language compiler mechanism for converting a multithreaded source program described by a program language into a computer executable machine language from a computer system "; also, Figs. 17a-17b, [0246]-[0247], [0255]-[0261], the sequentializer module consists of elements in 17a or 17b), and create as output a sequential program having assertions ([0255], "the actual race detection is carried. For this, 20 bytecodes; for instance, are instrumented which read or write to an object. Each time such a byte code is executed, it is checked whether it is a global object...If a global object is being dealt with, the extra data structures can be accessed and it can be verified, ...whether this instruction represents a data race. If so, this is flagged to the user", emphasis added).

Applicant argued:

Claim 1

Christiaens does describe the use of a compiler at paragraphs 0256-0261 on pages 13-14, but again, does not describe, as part of this compilation, the creation of any sequential program from a concurrent program.

Examiner's responses:

In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., use of a compiler, as part of this compilation, the creation of any sequential program from a concurrent program) are not recited in the

rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Even if applicant's arguments are not recited in the rejected claim 1, Christiaens still teaches compilation any sequential program from a concurrent program (Christiaens, [0034], "a computer language compiler mechanism for converting a multi-threaded source program described by a program language into a computer executable machine language from a computer system"). Christiaens teaches a compiler mechanism convert a multi-threaded source program into a computer executable machine language that is interpreted to create as output a sequential program.

Applicant argued:

Claim 13, Claim 18

Christiaens does not describe, teach, or suggest each and every element of amended claims 13 and 18.

Examiner's responses:

The examiner strongly disagree applicant's arguments. Christiaens teaches and suggests the limitation as recited in the amended claims 13 and 18, "creating a sequential program from the concurrent program, the sequential program containing assertions such that, during an analysis of the sequential program, when assertions are not met, the analysis of the sequential program indicates the presence of a data race in the concurrent program for the target variable" (Christiaens, Fig. 1, [0102], "The Java source code 101, once processed is intended to execute concurrently in a computer system 107 (CPU) as described above ... Java source code 101 can be compiled by a compiler 102 into class files 103, i.e. a type that defines the implementation of a particular kind of object, containing bytecodes ... The function of the monitor 111 is to produce a report 109 on concurrency state information concerning concurrently executing threads. The report may contain information on data races occurring while the class files 103

Art Unit: 2191

are executing in the system 107"; also, [0106], "for example, it two bytecodes modify a member variable of an object in the global set ... then a data race is reported", emphases added).

Christiaens teaches Java source code 101 can be compiled by compiler 102 into class files 103 class files that read on the limitation "creating a sequential program from the concurrent program. Moreover, Christiaens' report 109 read in the limitation "indicates the presence of a data race in the concurrent program for the target variable". Thus, Christiaens teaches all the limitations in Claims 13 and 18 of the present application.

### ***Conclusion***

10. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136 (a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Anna Deng whose telephone number is 571-272-5989. The examiner can normally be reached on Monday to Friday 9:30 AM - 6:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the TC2100 Group receptionist whose telephone number is 571-272-2100.

Art Unit: 2191

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Anna Deng

June 10, 2007

*a. d.*

A handwritten signature in black ink, appearing to read 'WZ', with a long, sweeping horizontal line extending to the right.

WEI ZHEN  
SUPERVISORY PATENT EXAMINER